INTELLIGENT ASSEMBLY THROUGH EXPERT SYSTEM-BASED PARTS DISTRIBUTION

Z. Katz* F. Zheng†

(First received September 1995; Final version February 1996)

Abstract

Programmable though intelligent robotic assembly tasks could be facilitated by the use of appropriate sensing and compatible expert systems. The concept analysis and integration of such an expert system is presented. A specific methodology allowing the use of the expert system for a multi-stage robotic assembly process is detailed. The incorporation of the expert system in a microcomputer controlled robotic assembly cell for intelligent assembly decisions is discussed.

Introduction

One obvious objective in the application of an intelligent system for the performance of automated assembly,[1] is to allow for a variety of sub-tasks to contribute to the extension of the flexibility of such a manufacturing system.[2]

The flow of parts for an intelligent assembly procedure requires the perfect identification and distribution of parts. As additional sub-assembly requirements may arise, same will apply to distribution and manipulation of the respective sub-assemblies.[3] The system is to be integrated and should perform according to the specific tasks required for the completion of the assembly.

Automated assembly operations vary from dedicated systems of rigid hardware such as feeders, orienting devices and stationary workheads to robotic assembly planning and operations in flexible manufacturing environments.[4] The use of robots in assembly operations requires specific methodologies emphasizing aspects of system modelling, knowledge-based data related to parts features,[5] desired sequence, sensory data, appropriate algorithms and local performance.

In this paper, the problem of integration of sequential and parallel assembly stations into a computer controlled robotic assembly cell, manipulating parts delivered in random order, is presented. An algorithm is proposed to resolve the problem. This is based on an expert system[6,7] conforming to a pre-determined set of rules. The result is an intelligent decision for the parts and sub-assemblies distribution and use of the system's multiple assembly stations.

Parts initially identified or sub-assembled are distributed in a manner accomplishing efficient use of the assembly cell.[8,9] The complexity of the problem is a result of a large number of combinations of possible assembly tasks related to a variety of parts and assembly stations. The work presents the expert system's structure, logic and application. It results in an intelligent decision of task performance by the assembly robot.

Problem description

An assembly robot-centred cell is dedicated to a multi-stage assembly operation of parts fed to the cell in a random order. The parts require identification, manipulation, and distribution to sub-assembly stations or buffer areas. The sub-assemblies are handled in a similar manner as the parts, while the complete system is based on intelligent decisions required at each stage of the assembly. The intelligent and flexible system has a knowledge base while containing information acquired by an appropriate sensory system. The assembly robot operations are well-defined and pre-programmed with input-output signals flow for timeously enabling the sub-routines required for the different process stages. However, the optimization of the whole assembly process in the sense of maximizing the use of the assembly hardware, is based on an expert system. Its purpose is to provide the intelligent decisions based on data related to the instantaneous conditions in the assembly cell according to a set of predetermined conditional rules.

The problem is complex. A solution requires data acquisition and processing, a system algorithm, hardware integration, software for the expert system and correct manipulation of parts during all assembly stages. There are specific details to be dealt with. Suppose there are n number of parts, i number of assembly stations and r number of buffer areas. The number of possible combinations of sub-tasks for the whole assembly process could be large. However, since the proposed approach aims at

*Professor, Rand Afrikaans University, Member
†Graduate student, Rand Afrikaans University

utilizing efficiently the system, it is not strongly dependent on those numbers. Furthermore, if the number of robots in the cell will increase, it will surely facilitate the process towards the achievement of the established goal. The dynamic changes, which are to an extent beyond the capacity of the cell to practically handle them efficiently, might have to be limited. The present proposed solution deals entirely with the possibility of intelligent, intermediate decisions and is based on the concept of fixed hardware but flexible manipulation of data. This is possible if the knowledge base is suitably designed for the cell operation.

The proposed incorporation of the expert system ensures a smooth assembly process according to an effective assembly process algorithm. The above formulated problem demands from the expert system the capability of accepting real-time data and output commands for the performance of the assembly by the respective cell structure. The concept of the proposed expert system is presented in Figure 1.

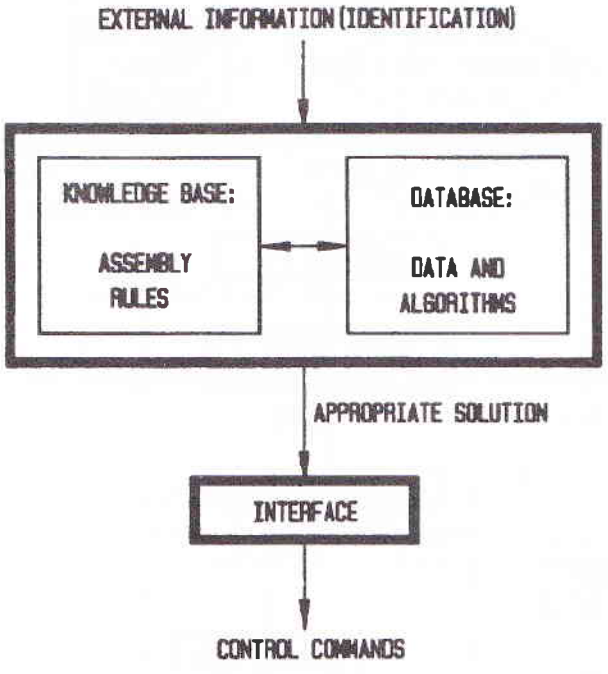


Figure 1 The expert system concept

Parts manipulation for routing

Parts manipulation to and from assembly stations is a complex problem considering the need for data generation, processing and distribution. This, coupled with hardware integration, may become impractical if optimization of time or parts scheduling becomes unavoidable.

We suggest here that the routing and therefore the parts manipulation for final assembly purpose be performed, though concurrently at times, in a manner in which the assembly cell will be efficient. This holistic approach will ensure that parts entering the cell are identified and handled logically according to the suggested expert system. It is assumed that parts could be, at any instant, either manipulated or assembled including temporary storage at buffer stations.

Figure 2 illustrates the software logic for an experimental assembly as performed in this work. It indicates the possibilities as could be experienced by the intelligent decisions of the expert systems incorporated in the assembly process software. The parts routing is, of course, defined according to certain basic rules related to assembly sequence or location.

The assembly domain is defined, the dynamic changes are recognized as a part of the assembly process and the algorithm represents knowledge-based data and application of predetermined set of rules. The diagnostic rule for a certain assembly procedure must contain all the necessary conditions for the occurrence of the event. The supervisory cell controller acts as coordinator and processes the sensory data for a smooth, though predetermined, sequence of assembly steps. Parts entering the system are so routed to exit as the final product.

The assembly matrix

The handling of a part by the system (robot or other hardware) under the cell controller supervision, is performed prior, during, and after an assembly stage. The handling is described and controlled by a matrix structure. It takes into consideration the specific group of assembly stations, the parts involved and a certain sequence of rules which together build up the expert system for the assembly. Figure 3 presents typical assembly stages as performed according to the suggested expert system.

A general matrix structure has the form:

$$[M] = [p] \cdot [k]$$

where:

- M = indicator of preference for selection of an assembly station to perform the particular assembly operation
- p = represents information to the computer indicating the nature of the assembled parts as output from a certain assembly station (see additional explanation)
- k = is an assembly performance factor

The above general matrix applies to each assembly stage as required by the specific case. For such a stage,

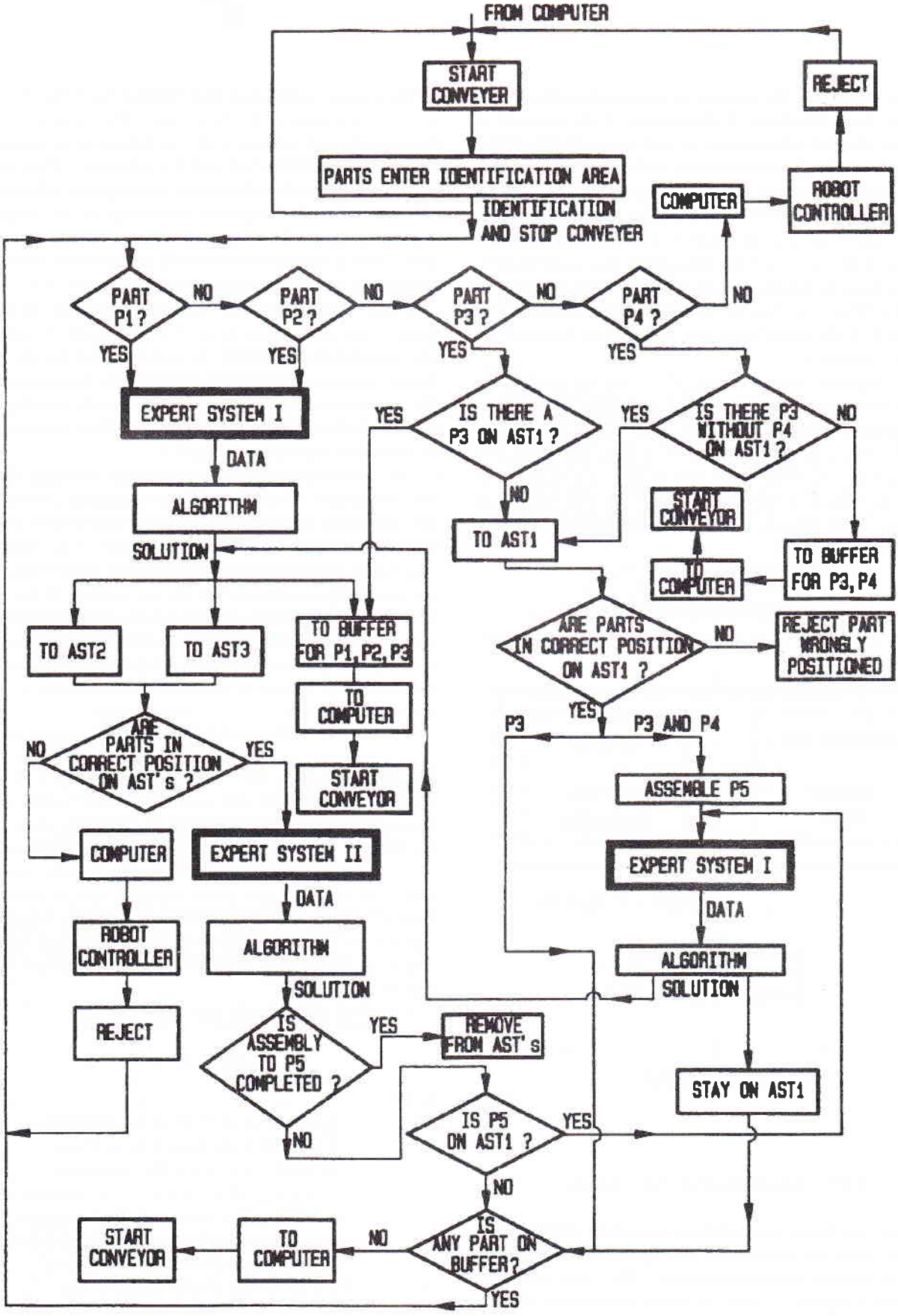


Figure 2 Software structure for assembly operation

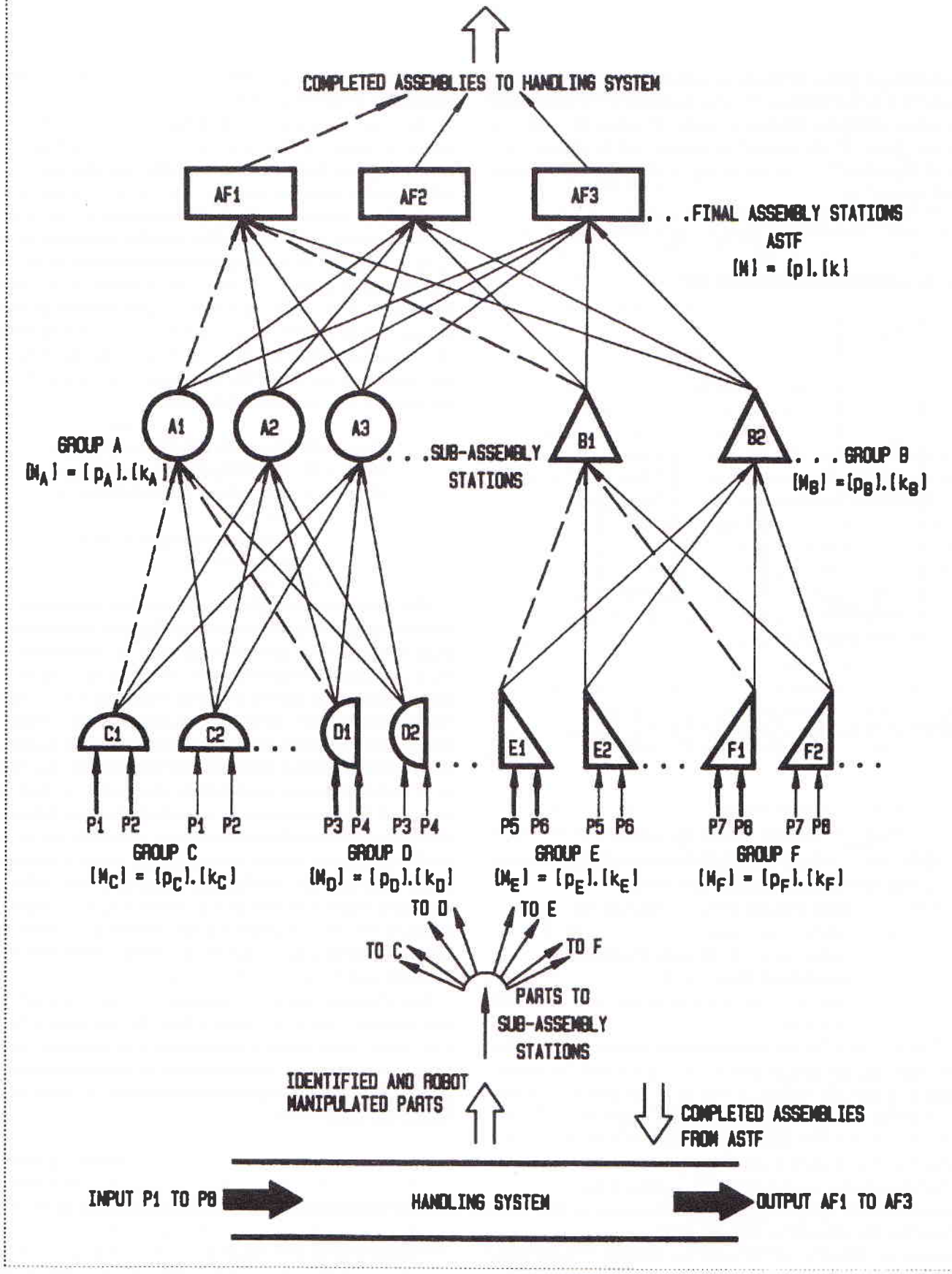


Figure 3 Assembly stages using expert system

a dedicated group of identical assembly stations is available. If n is the number of parts required to be assembled at every assembly station in each of i assembly stations in say group A, the respective matrix can be represented as in equation (1). For say group A, the matrix structure will appear as:

$$[M_A] = [p_A] \cdot [k_A]$$

or in a detailed format as shown:

$$\begin{bmatrix} M_{A1} \\ . \\ . \\ . \\ . \\ M_{Anxi} \end{bmatrix} = \begin{bmatrix} \alpha_{A1} \cdots\cdots \\ .\alpha_{A1} \cdots\cdots \\ \cdots\cdots\cdots \\ \cdots\cdots\cdots \\ \cdots\cdots\alpha_{A2}\cdots \\ \cdots\cdots\cdots\alpha_{Ai} \end{bmatrix} X \begin{bmatrix} k_{A1} \\ k_{A2} \\ . \\ . \\ . \\ k_{Anxi} \end{bmatrix}$$

$\leftarrow n$ element $\rightarrow$

$\leftarrow nxi$ element $\rightarrow$

(1)

where:
$$\alpha_{A1} = P_{u-v}A1 + P_{w-x}A1 + \ldots$$
$$\alpha = P_{u-v}A2 + P_{w-x}A2 + \ldots$$
$$\alpha_{Anxi} = P_{u-v}Ai + P_{w-x}Ai + \ldots$$

Here, the computer will read the following information in a standard format used in this work:

- P : part (subassembly or individual)
- $u - v$: parts u to v assembled in sequential order $(u < v)$. If individual part is considered then $v = 0$.
- Ai : assembly station i in group A of identical stations

The solution for the above matrix equation will define the type of part manipulation. It could either be assembled at a specific assembly station or temporarily placed at a buffer location. The conceptual structure of the expert system for the above and its software structure, are illustrated in Figure 4 and Figure 5, respectively.

Although the proposed expert system relies on the operation of the system controller according to the solutions for the respective matrices, it is imperative for the system to follow a set of rules as defined. Those rules are based on assembly logic and if followed, the solution could apply to a large number of possible combinations of assembly stations, buffer zones, parts and practical sequential order of assembly.

The expert system on which the concept of assembly stages is based, is schematically described in Figure 3. Here the parts handled and identified are distributed to sub-assembly stations and the process is propagated up to the final assembly. Each group requires the appropriate solution of the respective matrix according to the rules. This will output operational commands for the part handling system to finally incorporate, in a correct sequential order, all the assembly stages for each individual part. The system seeks values for p and k to assess the numerical value for M according to the structure of the matrix in eq. (1). The maximum values for M are decisive for the manipulation stage.

$M_{\max} > 0$	the part will go to assembly
$M_{\max} < 0$	the part will go to buffer
$M_{\max} = 0$	dependent type of part, to buffer independent type of part, to assembly

The rules for the k factor are explicit and based on assembly logic such as sequence of parts assembly and parts availability at the assembly stations. Therefore factor k is related to the sequence of assembly. For $k = 1$ the part could be assembled. Subsequently $k = -1$ indicates that the part would be out of a specific assembly station for a waiting period. A more detailed explanation and conditional rules for the decision on the value of k in each case, is included in Appendix A. For the values of k, it is important to define the two following conditions: (i) which part has been presently identified by the system and simultaneously, (ii) what is the situation regarding the availability of parts at the different assembly stations in the group. An example of required k values follows. In Figure 3 the assembly stations $C1$ and $C2$ accept parts P_1 and P_2 and output subassemblies $P_{1-2}C1$ and $P_{1-2}C2$, respectively.

The identical assembly stations $C1$ and $C2$ allow for two different parts $P1$ and $P2$ to be assembled here. The parts could be of a dependent (its assembly is related to other parts manipulation or assembly) and of an independent (its assembly not related to other parts assembly) type.

1. If $P_1C1 = 0$ (part 1 is **not** at sub-assembly station $C1$), then $k_{C1} = 1$: part P_1 could be assembled at $C1$ and should be directed there. At the same time, $k_{C2} = -1; k_{C4} = -1$. Alternatively, if $P_1C1 = 1$ (part 1 **is** at $C1$), then $k_{C1} = k_{C2} = k_{C4} = -1$ and the part P_1 could not be assembled at $C1$ at this stage.

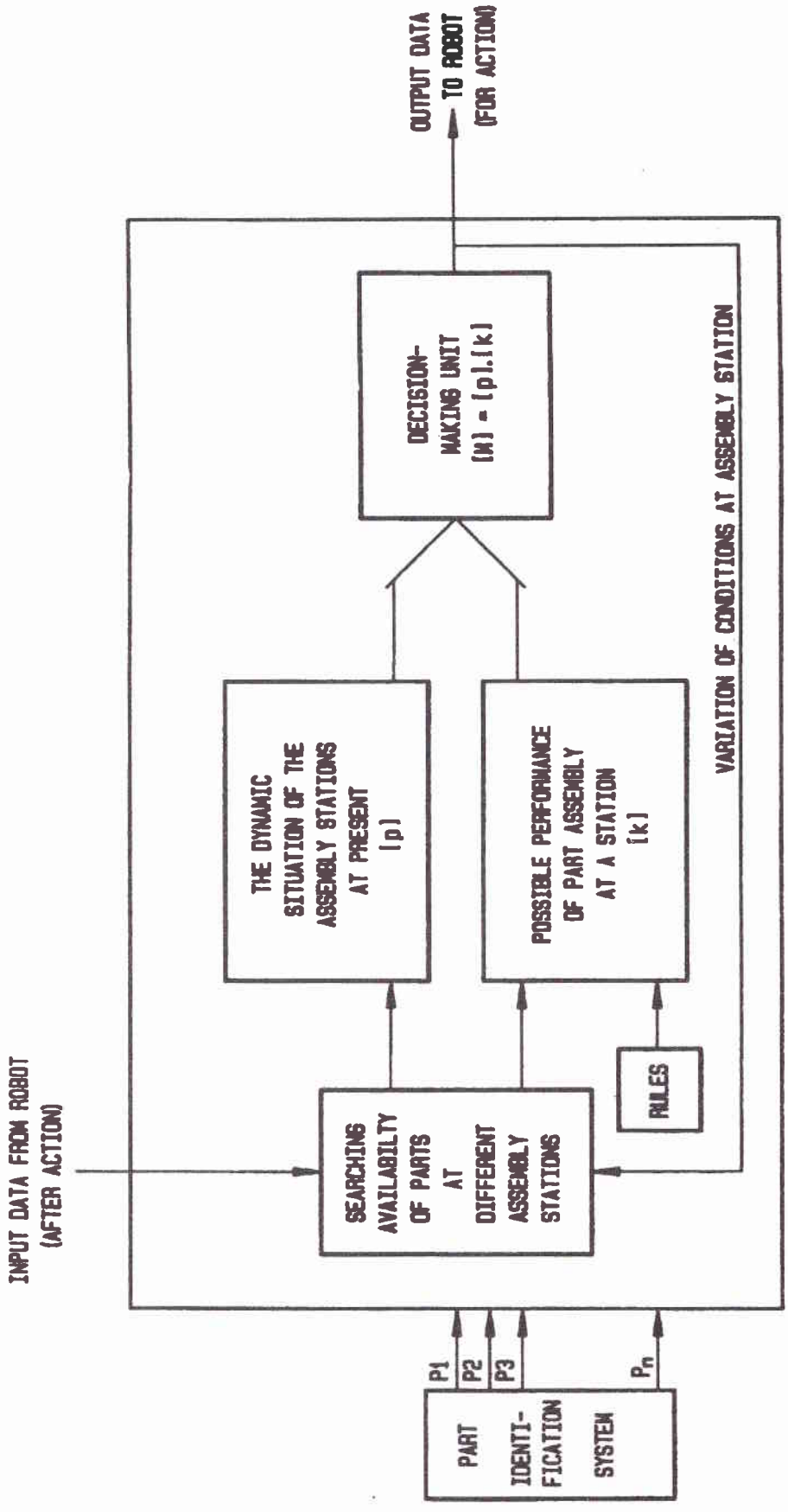


Figure 4 Conceptual structure of the expert system software

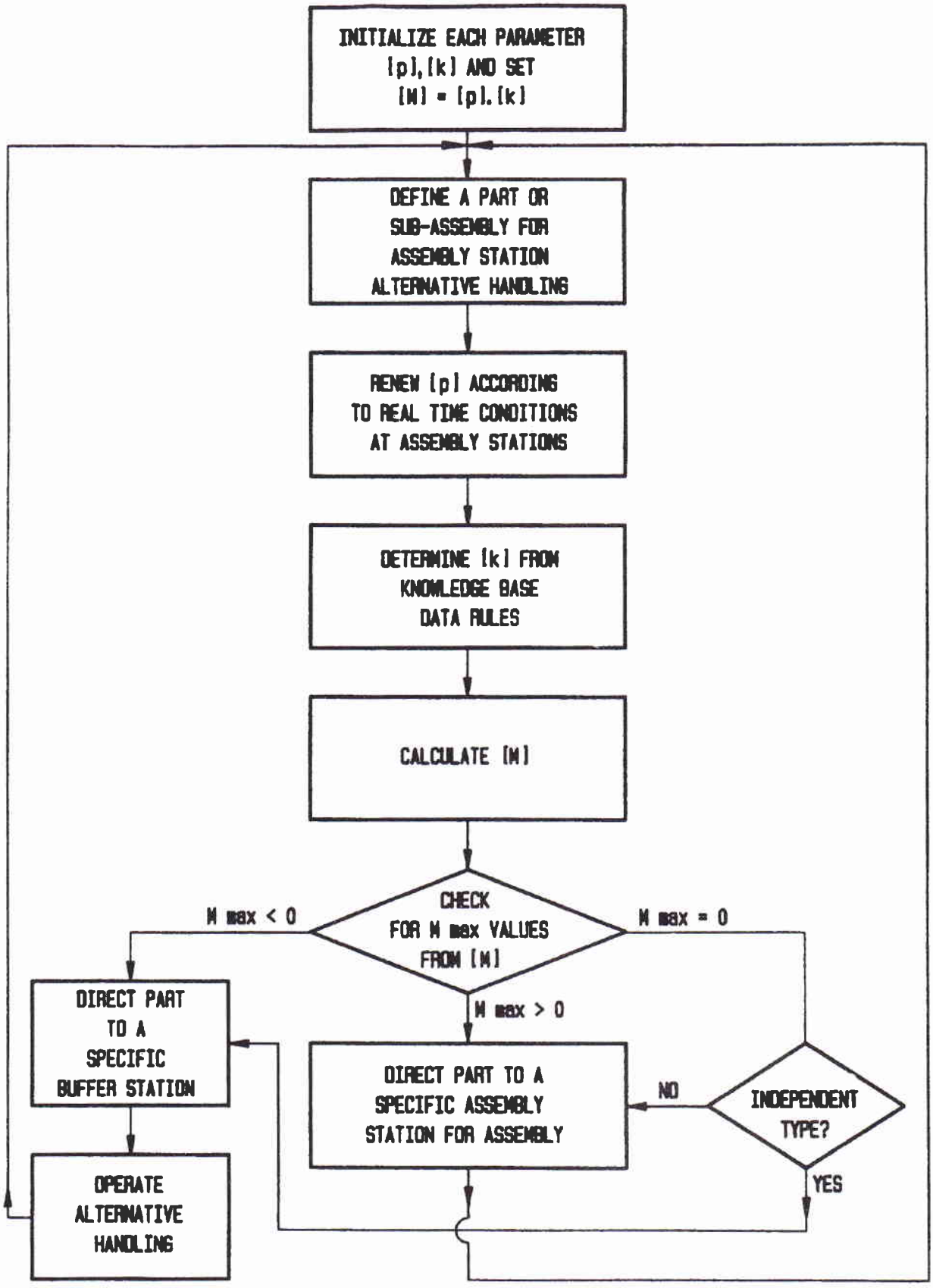


Figure 5 The software structure for the expert system

2. If $P_1C2 = 0$, the following occurs:

   $k_{C3} = 1$, $k_{C2} = -1$, $k_{C4} = -1$, alternatively, $k_{C2} = k_{C3} = k_{C4} = -1$. Assume P_2 is (a dependent part) subsequently identified. The following is possible:

3. If $P_2C1 = 0$ and $P_2C1 = 1$ then $k_{C2} = 1$ and $k_{C1} = k_{C3} = -1$. Alternatively, $P_2C1 = 1$ and $P_1C1 = 0$, then $k_{C1} = k_{C2} = k_{C3} = -1$.

4. If $P_2C2 = 0$ and $P_1C2 = 1$, then $k_{C4} = 1$ and $k_{C1} = k_{C3} = -1$. Alternatively, $k_{C4} = k_{C1} = k_{C3} = -1$.

Let us work out the values needed for the k matrix. According to the two conditions required as described previously:

1. Presently, the system identifies part P_1 and simultaneously,

2. the situation regarding availability of parts of the different assembly stations in the group C is $P_1C1 = 1$; $P_1C2 = 0$; $P_2C1 = 0$ and $P_2C2 = 0$.

The respective k_C values would be:
$k_{C1} = -1$; $k_{C2} = -1$; $k_{C3} = 1$ and $k_{C4} = -1$.
The final k matrix structure would be

$$[k_C] = \begin{bmatrix} k_{C1} \\ k_{C2} \\ k_{C3} \\ k_{C4} \end{bmatrix} = \begin{bmatrix} -1 \\ -1 \\ 1 \\ -1 \end{bmatrix}$$

For the same assembly stations group, the part matrix requires the values for

$$[P_C] = \begin{bmatrix} \alpha_{C1} & . & . & 0 \\ . & \alpha_{C1} & . & . \\ . & . & \alpha_{C2} & . \\ 0 & . & . & \alpha_{C2} \end{bmatrix}$$

here:

$$a_{C1} = P_{u-v}C1 + P_{w-x}C1 \ldots.$$
$$a_{C2} = P_{u-v}C2 + P_{w-x}C2 \ldots.$$

in this case:

$$a_{C1} = P_1C1 + P_2C1$$
$$a_{C2} = P_1C2 + P_2C2$$

and the matrix structure is:

$$[P_C] = \begin{bmatrix} P_1C1 + P_2C1 & 0 & 0 & 0 \\ 0 & P_1C1 + P_2C1 & 0 & 0 \\ 0 & 0 & P_1C2 + P_2C2 & 0 \\ 0 & 0 & 0 & P_1C2 + P_2C2 \end{bmatrix}$$

but for a case where at present, prior to identification of the next part:
$P_1C1 = 1$; $P_1C2 = 0$ and $P_2C1 = 0$; $P_2C2 = 0$.
In such a case the matrix is:

$$[P_C] = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

At this stage it is possible for the expert system to indicate the preference for the selection of the assembly station in the C group to perform the assembly. $[M_C] = [p_C] . [k_C]$ is now becoming, for the conditions stated:

$$\begin{bmatrix} M_{C1} \\ M_{C2} \\ M_{C3} \\ M_{C4} \end{bmatrix} = \begin{bmatrix} 1\,0\,0\,0 \\ 0\,1\,0\,0 \\ 0\,0\,0\,0 \\ 0\,0\,0\,0 \end{bmatrix} \begin{bmatrix} -1 \\ -1 \\ 1 \\ -1 \end{bmatrix} = \begin{bmatrix} -1 \\ -1 \\ 0 \\ 0 \end{bmatrix}$$

It is evident that for a P_1, as identified, and of an independent type and $M_{\max} = M_{C3} = 0$, the expert system will resolve to direct P_1 to the assembly station $C2$. Should P_1 have been a dependent type of part, it would have been directed to the buffer station. An additional look at Figure 3 would by now clarify the possible expansion of the expert system to include several parts, different groups of assembly stations and several such stations in a group. The handling system, the manipulating robot in this case, is therefore now capable of distributing correctly the identified parts to initial assembly stations and in a similar way proceeding with the manipulation of the sub-assembly up to the final assembly. The expert system is capable now of replacing the initial required parts identification system with the correct decision to direct the sub-assembly to the next stage.

The system software and hardware

The assembly process performance requires a decision expected from the proposed expert system. It would select the instantaneous preference of a certain assembly station, related to the identification of a part/subassembly and reference signals from the manipulator. The concept of the software structure as presented in Figure 4 is based on the following practical sequence of events. Pending the part or sub-assembly identification, the cell controller uses its database to rename the parts in real time obtaining the $M_{\max}$ value

needed for the robot action. This is performed by solving the respective matrix equations $[M] = [p].[k]$. According to the previously explained set of predetermined rules related to the part identification, sequence of assembly operations and real-time conditions at assembly stations. It is obvious that the software is integrated and combined with the cell controller logical action related to a variety of activities within the whole robotic assembly cell. Figure 5 represents a typical software structure for the cell action, in which the specific expert system algorithms and respective solutions, are incorporated. In this case the system's hardware was tested for a specific type of parts assembly. The system's hardware consists of assembly stations, buffer stations, assembly robot, cell controller (IBM-286), sensors, alternative handling system and a dedicated parts identification system.

Summary

The concept incorporated into the proposed expert system for intelligent assembly has been applied to a specific robotic assembly cell. Identification of randomly delivered parts to the cell has been performed satisfactorily. The distribution of parts to first-line of assembly stations and further on towards the final assembly stage was accomplished by the assembly robot following the algorithm as related to the respective expert system based on the presented assembly matrix.

The performance could be extended in principle to a multi-robot cell. It would be of interest to investigate such a case coupled with a variation of parts structure and assembly requirements. It is possible to refine the methodology presented by incorporating an even more extended knowledge-based data related to specific distribution and assembly requirements.

References

1. Boothroyd G, Poli C & Murch L 1982. *Automatic Assembly.* Marcel Dekker Inc.

2. Huang KI. 1993. Planning Robotic Assembly for FMS. *Proc. Int. Symposium on Intelligent Robotics*, pp. 583–592.

3. Lee S & Shin YG 1990. Assembly planning based on sub-assembly extraction. *Proc IEEE Int Conf robotics and automation*, pp. 1606–1611.

4. Ranky GP 1983. *The design and operation of FMS.* IFS Publications, U.K.

5. Tonsoff HK, Menzel E & Park HS 1992. Knowledge based system for automated assembly planning, *CIRP Annals*, **41**, pp, 19–24.

6. Kumera SRT 1989. *Expert data base system in manufacturing.* Ind Eng and Management Press.

7. Fox J 1984. Expert systems and concept of knowledge. *Robotics and Artificial Intelligence*, Springer-Verlag, pp. 593–610.

8. Giusti F, Santochi M & Dini G 1991. Robotized assembly of modular fixtures. *CIRP Annals*, **40**, pp. 17–20.

9. Homen de Mello LS & Sanderson AC 1991. Representation of mechanical assembly sequences. *IEEE Trans on Robotics and Automation*, **6**, pp. 188–189.

Acknowledgment

The authors acknowledge the support of the Foundation for Research Development through its research grant for Manufacturing Systems Research.

Appendix A

Rules and definitions related to the determination of k values required for the assembly performance are detailed. According to Figure 3 in the text:

n	number of parts to be assembled
i	number of assembly stations in the respective group
P_1	independent type of part (or subassembly (i)
P_2	dependent type of part (or subassembly (d)
P_2	(dependent on P_1 for assembly performance)

The matrix structure:

$$[k] = \begin{bmatrix} k_1 \\ k_2 \\ . \\ . \\ k_{\text{nxi}} \end{bmatrix}$$

(i) P_1 type of part is identified or available.

rule (a): If $P_1C1 = 0$ then $k_{C1} = 1$. All other k values for the matrix are -1 except k_{Cn+1}; k_{C2n+1}; $k_{C3n+1} \ldots k_{Cmn+1}$ (here $mn+1 \leq ni$) which are not determined by this rule only. Otherwise, k values are -1 except k_{Cn+1}; k_{C2n+1}; $\ldots$ k_{Cmn+1} which are not determined by this rule only.

rule (b): If $P_1C2 = 0$ then $k_{Cn+1} = 1$. All other k values are -1 except k_{C1}; $k_{C2n+1;\, k_{C3n+1}} \cdots k_{Cmn+1}$. Otherwise, k values are -1 except

k_1; k_{C2n+1}; ... k_{Cmn+1} which are not determined by this rule only.

rule (i)$_i$: If P_1 Ci = 0 then $k_{Cmn+1} = 1$. All other values are -1 except k_{C1}; k_{Cn+1}; k_{C2n+1}... which are not (exclude k_{Cmn+1}) determined by this rule only.

(ii) P_2 type of part is identified or available.

rule (A): If $P_2C1 = 0$ and $P_1C1 = 1$, then $kC2 = 1$. All other k values are -1 except k_{Cn+2}; k_{C2n+2} ... k_{Cmn+2} which are not determined by this rule only. Otherwise k values are -1 except k_{Cn+2}; k_{C2n+2} ... k_{Cmn+2} which are not determined by this rule only.

rule (B): If $P_2C2 = 0$ and $P_1C2 = 1$, then $k_{Cn+2} = 1$. All other k values are -1 except k_{C2}; k_{C2n+2} ... k_{Cmn+2}. Otherwise k values are -1 except k_{C2}; k_{C2n+2}; k_{C3n+2}... which are not determined by this rule only.

rule (i)$_d$: If P_2 Ci = 0 and P_1 Ci = 1, then $k_{Cmn+2} = 1$. All other k values are -1 except k_{C2}; k_{C2n+2} ... which are not determined by this rule only. Otherwise k values are -1 except k_{C2}; k_{C2m+2} ... which are not determined by this rule alone.

Comments

1. If P_1 and P_2 are of an independent type, they are interchangeable within the respective set of rules.
2. Any part P_n will be classified as either dependent or independent type and be referred to as P_1 or P_2, respectively, and the rules will apply.